

# 1.创建Vue

---

## 1.下载

## 2.安装

## 3.添加环境变量

3.1 进入环境变量，编辑【系统变量】下的变量【Path】

3.2 添加Node.js的安装路径（此处为E:\devTools\nodejs\）

## 4.验证是否安装成功

## 5、修改模块下载位置

1.查看npm默认存放位置

2.在 nodejs 安装目录下，创建 “node\_global” 和 “node\_cache” 两个文件夹

## 6.通过NPM安装Vue

## 7.0安装webpack及webpack-cli

1.安装webpack

2.安装webpack-cli

3.检查是否安装成功

## 8.新建一个默认的Vue项目

1.创建项目

2.启动项目

# 1.下载

<http://nodejs.cn/download/>

## 下载

长期支持版本: **16.16.0**

<b>16.16.0</b> 长期支持版本	<b>18.7.0</b> 最新版本	
 <b>Windows 安装包</b> node-v16.16.0-x64.msi	 <b>macOS 安装包</b> node-v16.16.0.pkg	 <b>源代码</b> node-v16.16.0.tar.gz

**Windows 安装包 (.msi)**

**Windows 二进制文件 (.zip)**

**macOS 安装包 (.pkg)**

**macOS 二进制文件 (.tar.gz)**

**Linux 二进制文件 (x64)**

**Linux 二进制文件 (ARM)**

**Docker 镜像**

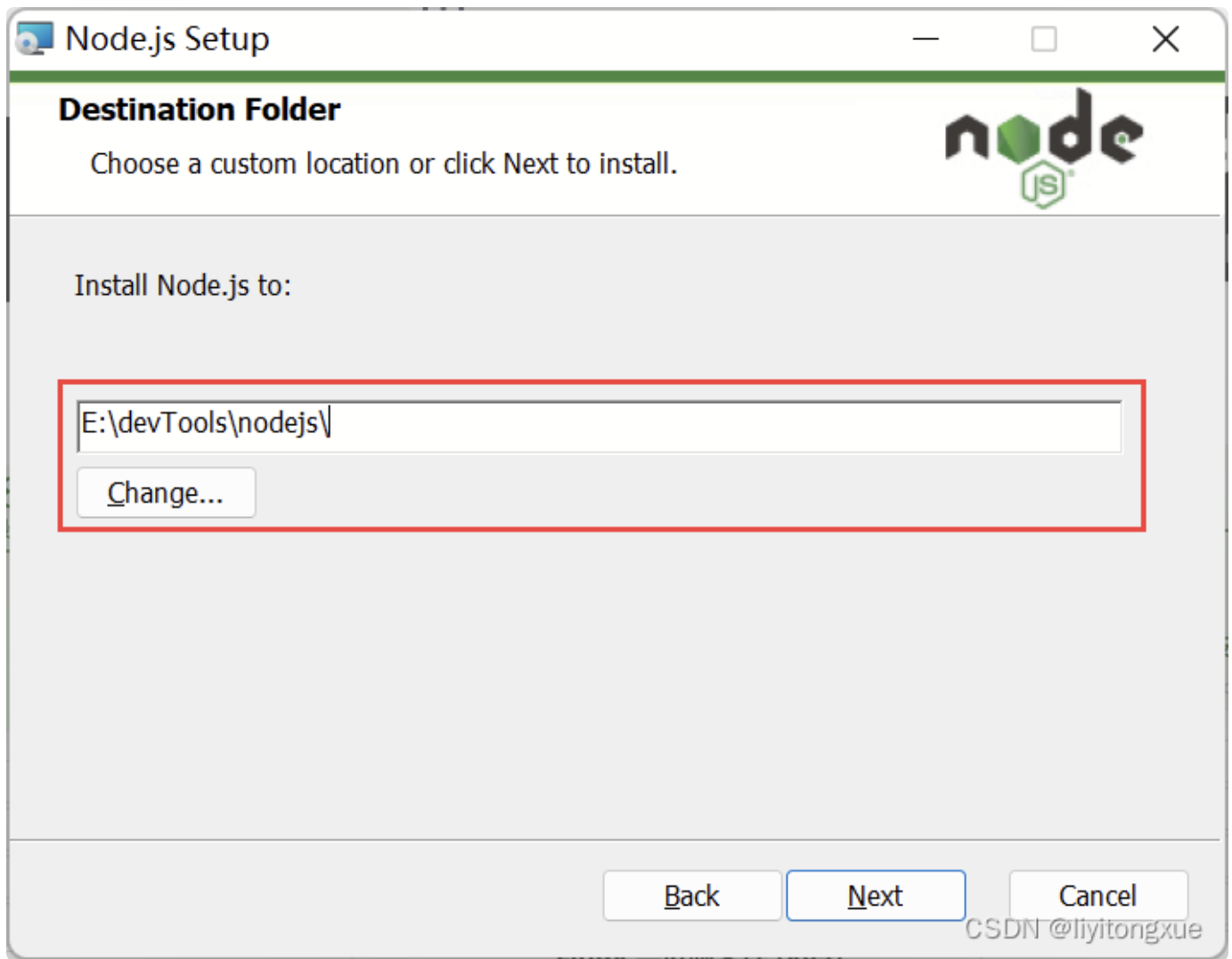
**全部安装包**

32 位	64 位
32 位	64 位
64 位 / ARM64	
64 位	ARM64
64 位	
ARMv7	ARMv8
官方镜像	
全部安装包	

- 发布文件的签名 SHASUMS (如何验证)
- 所有下载选项
- [通过包管理器安装 Node.js](#)

## 2. 安装

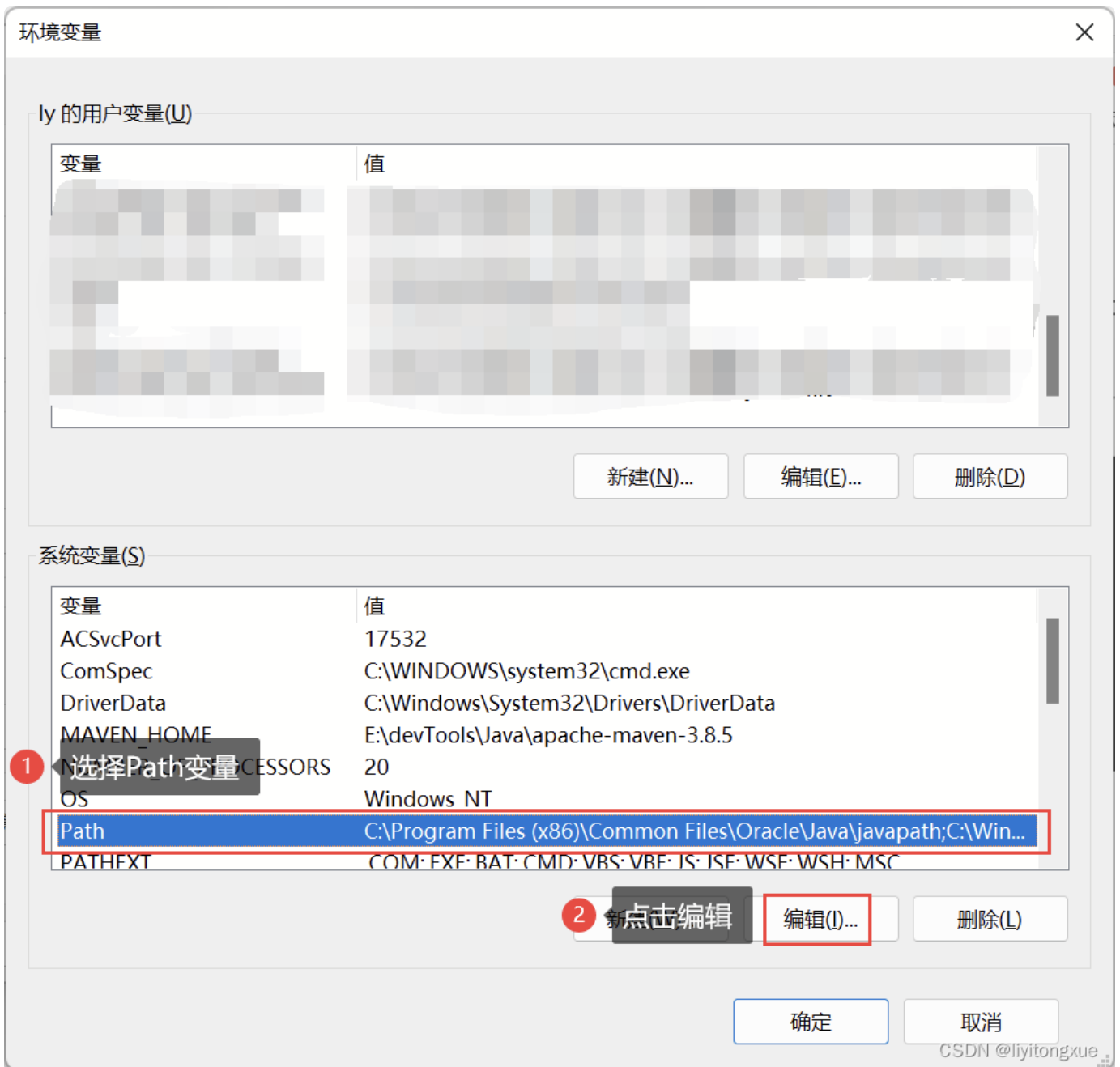
双击安装包，点击Next，勾选使用许可协议，点击Next，选择安装位置（可根据个人情况更换路径，我这里选择安装在E:\devTools\nodejs）



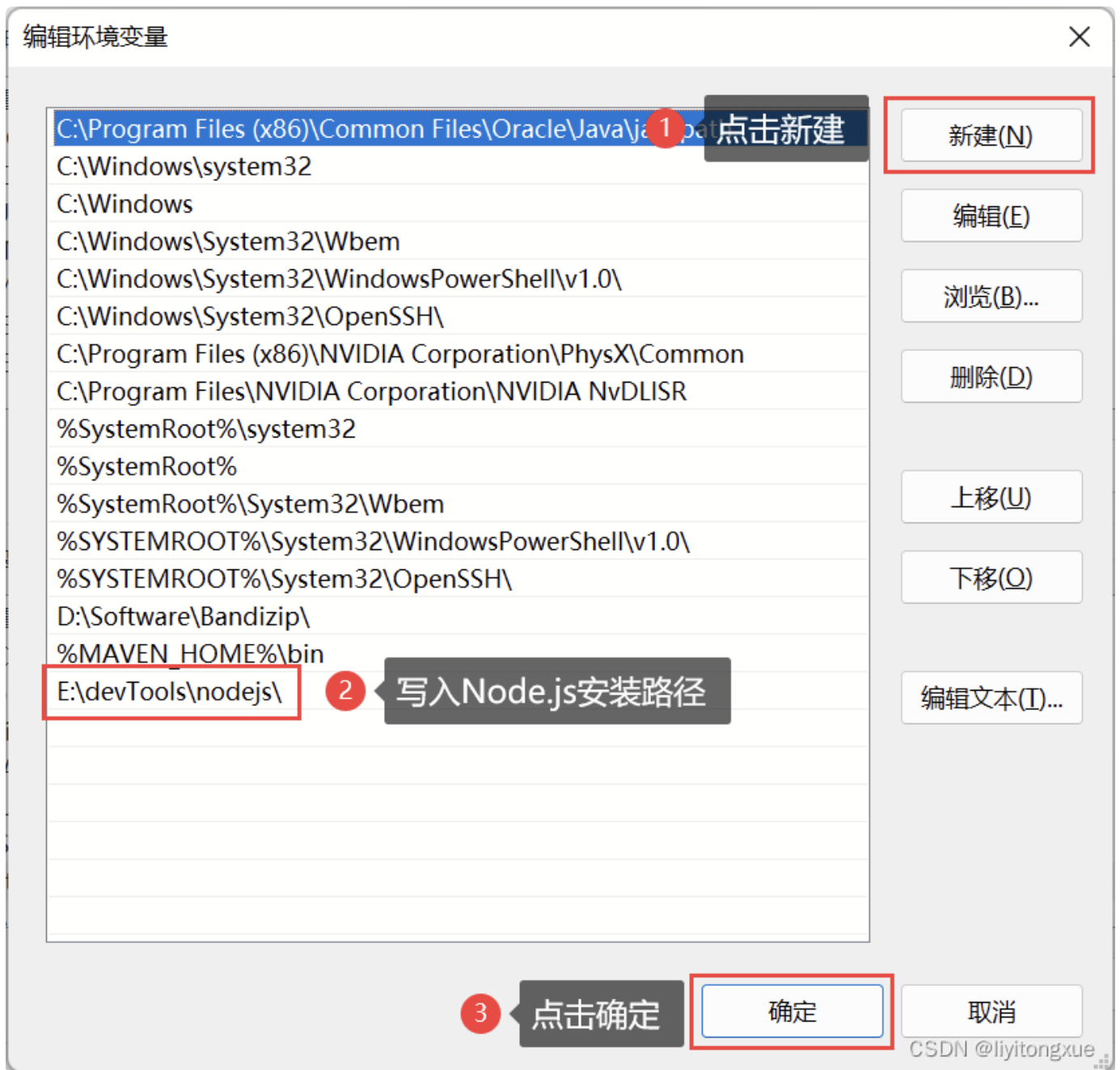
继续点击Next，点击Next，点击Install，点击Finish完成安装。

## 3.添加环境变量

### 3.1 进入环境变量，编辑【系统变量】下的变量【Path】



### 3.2 添加Node.js的安装路径（此处为E:\devTools\nodejs\）



## 4. 验证是否安装成功

进入cmd命令行窗口，输入node -v查看nodejs版本

```
XML | 复制代码
1 node -v
2 npm -v
```

如下图所示，即为安装成功：

C:\Windows\System32\cmd.exe

```
C:\Users\ly>node -v  
v16.15.0
```

```
C:\Users\ly>npm -v  
8.5.5
```

```
C:\Users\ly>
```

CSDN @liyitongxue

## 5、修改模块下载位置

此步骤修改以后npm全局下载模块的保存位置，可根据自身情况选择是否更改。

### 1.查看npm默认存放位置

- ▼ XML 复制代码
- 1 #使用npm get prefix查看npm全局模块的存放路径
  - 2 npm get prefix
  - 3 #使用npm get cache查看npm缓存默认存放路径
  - 4 npm get cache

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ly>npm get prefix
C:\Users\ly\AppData\Roaming\npm

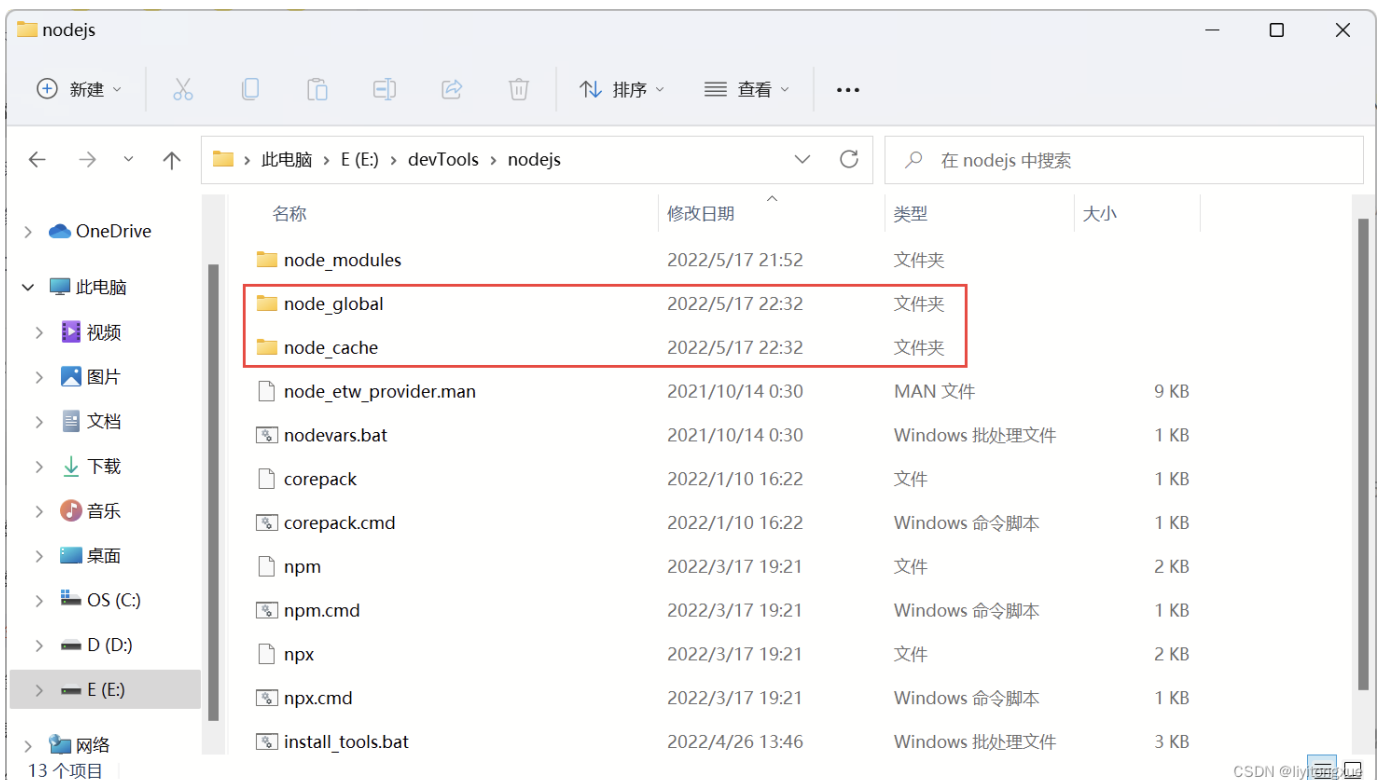
C:\Users\ly>npm get cache
C:\Users\ly\AppData\Local\npm-cache

C:\Users\ly>
```

CSDN @liytongxue

如上图所示，npm 全局模块存放位置以及cache的存放位置，默认是在 C 盘“C:\Users\用户\AppData”下。

## 2.在 nodejs 安装目录下，创建“node\_global”和“node\_cache”两个文件夹



## 3.修改默认文件夹

```
1  设置全局模块的安装路径到 “node_global” 文件夹,  
2  
3  npm config set prefix "E:\devTools\nodejs\node_global"  
4  
5  设置缓存到 “node_cache” 文件夹  
6  
7  npm config set cache "E:\devTools\nodejs\node_cache"
```

```
C:\WINDOWS\system32\cmd.exe  
Microsoft Windows [版本 10.0.22000.675]  
(c) Microsoft Corporation。保留所有权利。
```

```
C:\Users\ly>npm get prefix  
C:\Users\ly\AppData\Roaming\npm
```

```
C:\Users\ly>npm get cache  
C:\Users\ly\AppData\Local\npm-cache
```

```
C:\Users\ly>npm config set prefix "E:\devTools\nodejs\node_global"
```

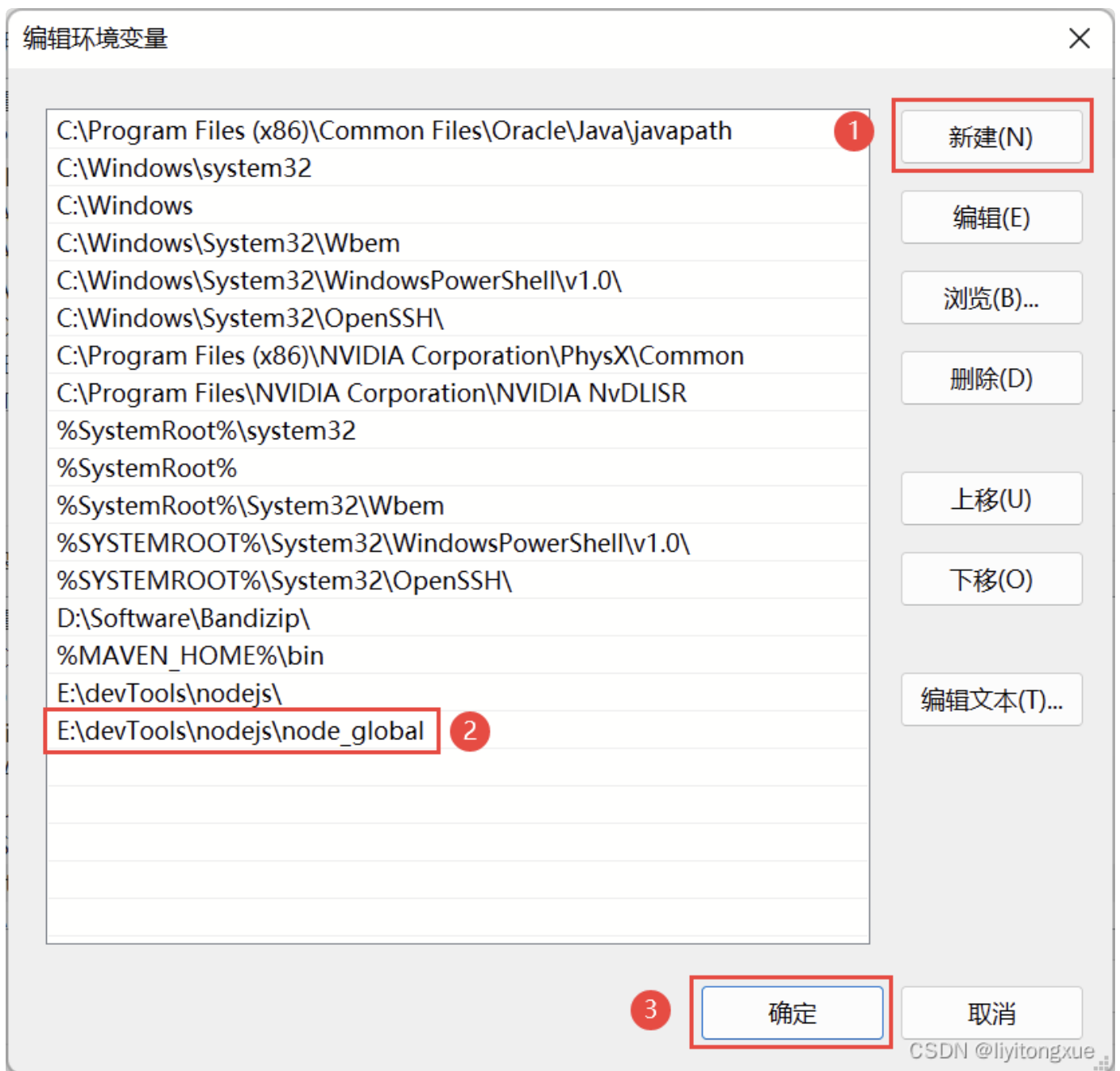
```
C:\Users\ly>npm config set cache "E:\devTools\nodejs\node_cache"
```

```
C:\Users\ly>
```

CSDN @liyitongxue

**注意：**由于 node 全局模块大多数都是可以通过命令行访问的，还要把【node\_global】的路径“E:\devTools\nodejs\node\_global”加入到【系统变量】下的【PATH】变量中，方便直接使用命令行运行，如下图所示：





#### 4.测试默认位置是否更改成功

经过上面的步骤，nodejs下载模块就会自动下载到我们自定义的目录，接下来我们测试一下是否更改成功。输入下面的命令：

```
1 npm install express -g
2 或者
3 npm install express --global
```

注意：“-g”等同于“-global”，“-g”是全局安装，不加“-g”就是默认下载到当前目录。“-g”表示安装到之前设置的【node\_global】目录下，同时nodejs会自动地在node\_global文件夹下创建

【node\_modules】子文件夹，即自动下载到“E:\devTools\nodejs\node\_global\node\_modules”路径下。

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ly>npm install express -g

added 57 packages, and audited 58 packages in 20s

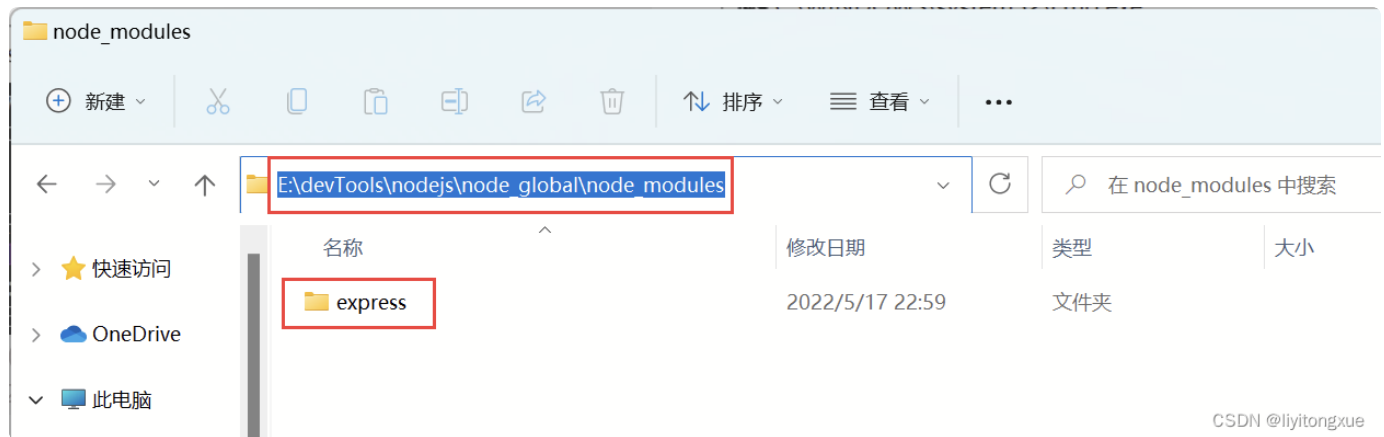
7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\ly>
```

CSDN @liyitongxue

如上图所示，下载express模块成功，然后在文件管理器中查看是否保存到上面自定义的路径下。



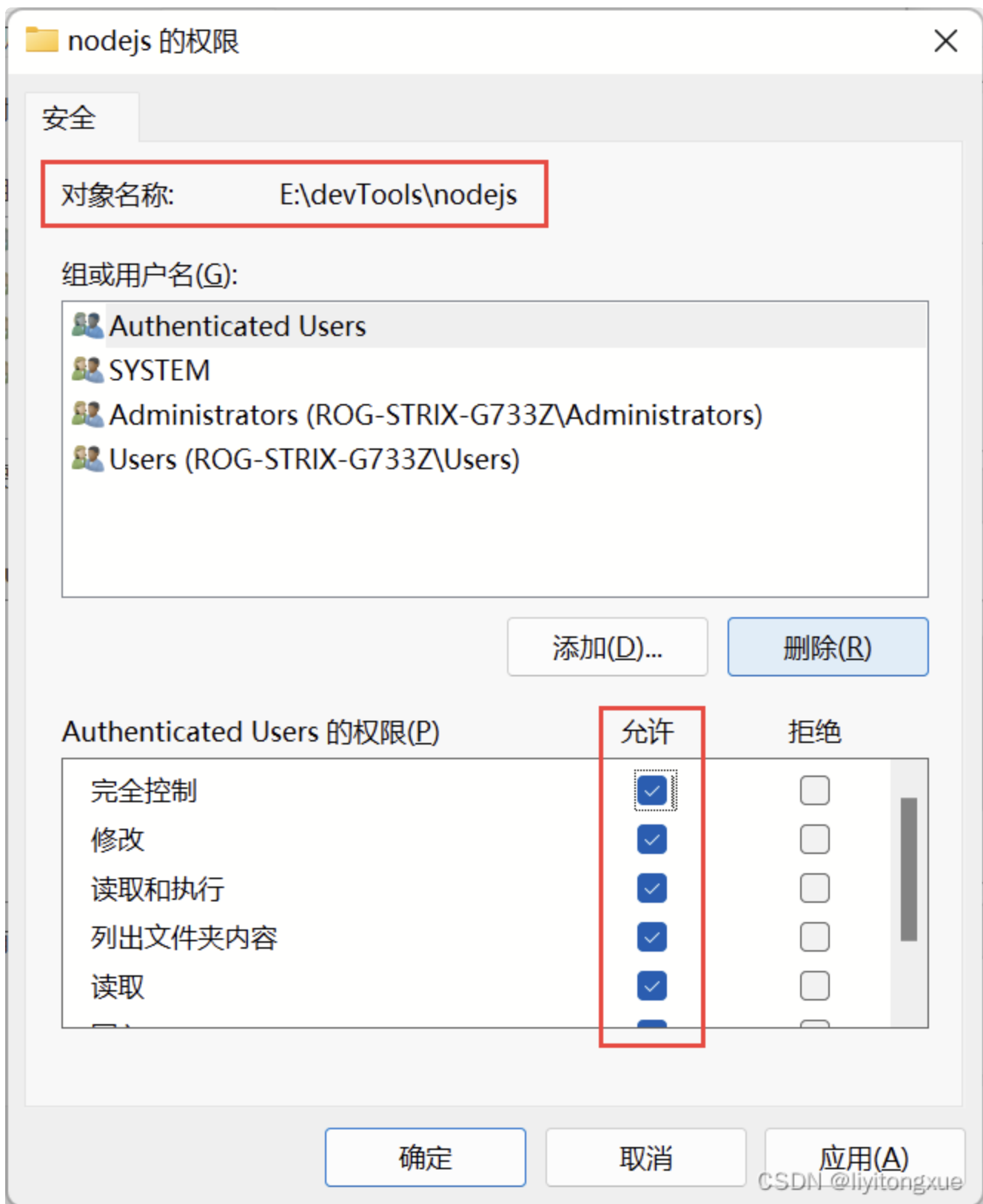
可以看到，express模块已经成功地下载到【E:\devTools\nodejs\node\_global\node\_modules】下。

注意：若执行命令npm install express -g出现如下报错：

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ly>npm install express -g
npm ERR! code EPERM
npm ERR! syscall mkdir
npm ERR! path E:\devTools\nodejs\node_cache\_cacache
npm ERR! errno EPERM
CSDN @liyitongxue
```

是由于对文件夹操作的权限不够，右击Nodejs文件夹->属性->安全，点击编辑，将所有权限都✓即可。



※执行npm install express -g仍然出错的话继续将nodejs下【node\_cache】、【node\_global】、【node\_modules】这三个文件夹的所有权限勾选，再次执行：

```
1 npm install express -g
```

即可下载成功。

## 6.通过NPM安装Vue

在Cmd命令行窗口中输入如下命令安装Vue脚手架。

```
1 npm install @vue/cli -g
```

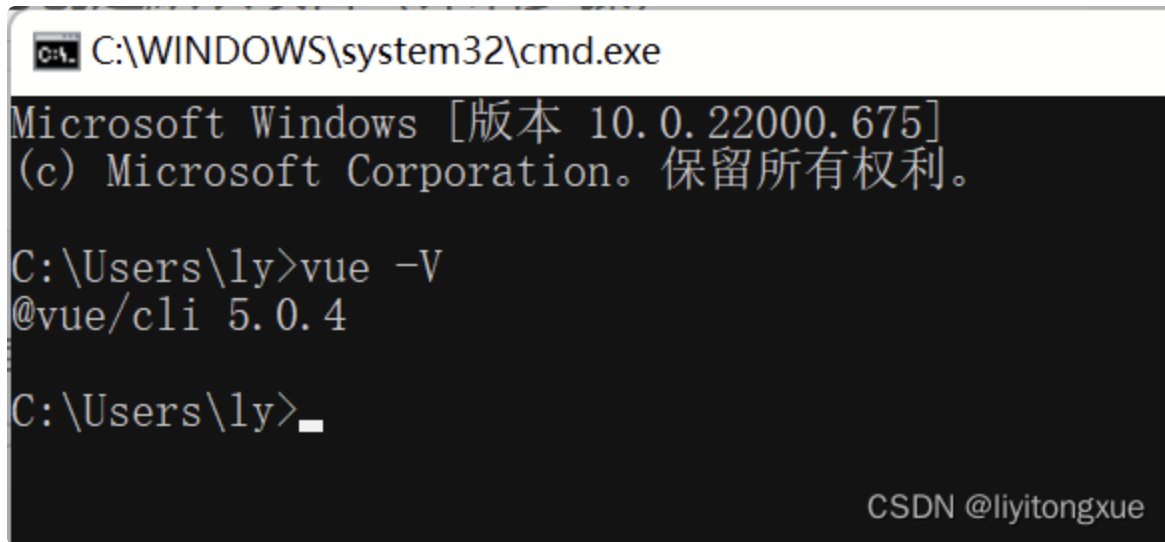
执行cnpm install @vue/cli -g后如下图所示：

```
C:\WINDOWS\system32\cmd.exe
deprecate @vue/cli-ui@5.0.4 > apollo-server-express@2.25.3 > graphql-tools@4.0.8 This package has been deprecated and now it only exports makeExecutableSchema. And it will no longer receive updates. We recommend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc. Check out https://www.graphql-tools.com to learn what package you should use instead
deprecate @vue/cli-ui@5.0.4 > apollo-server-express@2.25.3 > graphql-tools@4.0.8 > uuid@3.1.0 Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
deprecate vue-codemod@0.0.5 > jscodeshift@0.11.0 > micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.0 See https://github.com/lydell/source-map-resolve#deprecated
deprecate @vue/cli-ui@5.0.4 > apollo-server-express@2.25.3 > apollo-server-core@2.25.3 > apollo-cache-control@0.14.0 The functionality provided by the 'apollo-cache-control' package is built in to 'apollo-server-core' starting with Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#cachecontrol for details.
deprecate @vue/cli-ui@5.0.4 > apollo-server-express@2.25.3 > apollo-server-core@2.25.3 > apollo-tracing@0.15.0 The 'apollo-tracing' package is no longer part of Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#tracing for details
deprecate @vue/cli-ui@5.0.4 > apollo-server-express@2.25.3 > apollo-server-core@2.25.3 > graphql-extensions@0.15.0 The 'graphql-extensions' API has been removed from Apollo Server 3. Use the plugin API instead: https://www.apollographql.com/docs/apollo-server/integrations/plugins/
deprecate vue-codemod@0.0.5 > jscodeshift@0.11.0 > micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > resolve-url@0.2.1 https://github.com/lydell/resolve-url#deprecated
deprecate vue-codemod@0.0.5 > jscodeshift@0.11.0 > micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > uri-js@0.1.0 Please see https://github.com/lydell/urijs#deprecated
deprecate vue-codemod@0.0.5 > jscodeshift@0.11.0 > micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > source-map-url@0.4.0 See https://github.com/lydell/source-map-url#deprecated
All packages installed (798 packages installed from npm registry, used 8s(network 8s), speed 4.7MB/s, json 707(5.28MB), tarball 32.18MB, manifests cache hit 0, etag hit 0 / miss 0)
[@vue/cli@5.0.4] link E:\devTools\nodejs\node_global\vue@ -> E:\devTools\nodejs\node_global\node_modules\@vue\cli\bin\vue.js
C:\Users\ly>
```

cmd命令行窗口输入vue -V查看@vue/cli是否安装成功

```
1 vue -V
```

如下图所示，vue脚手架即安装成功



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ly>vue -V
@vue/cli 5.0.4

C:\Users\ly>_
```

CSDN @liyitongxue

## 7.0安装webpack及webpack-cli

### 1.安装webpack

由于webpack5及以上的版本变动较大，所以如果是采用vue3创建vue项目，用webpack4的版本更能互相的兼容。此处安装webpack@4.42.0版本，命令如下：

1    npm install webpack@4.42.0 -g

安装成功后，cmd命令行窗口如下图所示：

```
C:\WINDOWS\system32\cmd.exe
[14/23] webpack-sources@1.4.1 installed at node_modules\webpack-sources@1.4.3@webpack-sources
[15/23] memory-fs@0.4.1 installed at node_modules\memory-fs@0.4.1@memory-fs
[16/23] ajv@6.10.2 installed at node_modules\ajv@6.12.6@ajv
[17/23] @webassemblyjs/wasm-edit@1.8.5 installed at node_modules\@webassemblyjs_wasm-edit@1.8.5@@webassemblyjs\wasm-edit
[18/23] @webassemblyjs/wasm-parser@1.8.5 installed at node_modules\@webassemblyjs_wasm-parser@1.8.5@@webassemblyjs\wasm-parser
[19/23] @webassemblyjs/ast@1.8.5 installed at node_modules\@webassemblyjs_ast@1.8.5@@webassemblyjs\ast
[20/23] watchpack@1.6.0 installed at node_modules\watchpack@1.7.5@watchpack
[21/23] node-libs-browser@2.2.1 installed at node_modules\node-libs-browser@2.2.1@node-libs-browser
[22/23] micromatch@3.1.10 installed at node_modules\micromatch@3.1.10@micromatch
[23/23] terser-webpack-plugin@1.4.3 installed at node_modules\terser-webpack-plugin@1.4.5@terser-webpack-plugin
deprecate node-libs-browser@2.2.1 > url@0.11.0 > querystring@0.2.0 The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
deprecate watchpack@1.7.5 > watchpack-chokidar2@2.0.1 > chokidar@2.1.8 Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies
deprecate micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.0 See https://github.com/lydell/source-map-resolve#deprecated
deprecate micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > source-map-url@0.4.0 See https://github.com/lydell/source-map-url#deprecated
deprecate micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > resolve-url@0.2.1 https://github.com/lydell/resolve-url#deprecated
deprecate micromatch@3.1.10 > snapdragon@0.8.2 > source-map-resolve@0.5.3 > urix@0.1.0 Please see https://github.com/lydell/urix#deprecated
All packages installed (308 packages installed from npm registry, used 3s(network 2s), speed 2MB/s, json 271(1.15MB), tarball 3.75MB, manifests cache hit 0, etag hit 0 / miss 0)
[webpack@4.42.0] link E:\devTools\nodejs\node_global\webpack@ -> E:\devTools\nodejs\node_global\node_modules\webpack\bin\webpack.js
C:\Users\ly>
```

## 2.安装webpack-cli

另外：由于webpack的版本需要webpack-cli一起配合使用，执行如下命令安装webpack-cli：

XML | 复制代码

```
1  cnpm install webpack-cli -g
```

安装成功后，cmd命令行窗口如下图所示：

```
C:\WINDOWS\system32\cmd.exe
C:\Users\ly>cnpm install webpack-cli -g
Downloading webpack-cli to E:\devTools\nodejs\node_global\node_modules\webpack-cli_tmp
Copying E:\devTools\nodejs\node_global\node_modules\webpack-cli_tmp\webpack-cli@4.9.2@webpack-cli to E:\devTools\nodejs\node_global\node_modules\webpack-cli
Installing webpack-cli's dependencies to E:\devTools\nodejs\node_global\node_modules\webpack-cli\node_modules
[1/12] fastest-levenshtein@ 1.0.12 installed at node_modules\fastest-levenshtein@1.0.12@fastest-levenshtein
[2/12] interpret@ 2.2.0 installed at node_modules\interpret@2.2.0@interpret
[3/12] colorette@ 2.0.14 installed at node_modules\colorette@2.0.16@colorette
[4/12] @discoveryjs/json-ext@ 0.5.0 installed at node_modules\_@discoveryjs_json-ext@0.5.7@@discoveryjs\json-ext
[5/12] @webpack-cli/configtest@ 1.1.1 installed at node_modules\_@webpack-cli_configtest@1.1.1@@webpack-cli\configtest
[6/12] @webpack-cli/serve@ 1.6.1 installed at node_modules\_@webpack-cli_serve@1.6.1@@webpack-cli\serve
[7/12] commander@ 7.0.0 installed at node_modules\commander@7.2.0@commander
[8/12] @webpack-cli/info@ 1.4.1 installed at node_modules\_@webpack-cli_info@1.4.1@@webpack-cli\info
[9/12] execa@ 5.0.0 installed at node_modules\execa@5.1.1@execa
[10/12] webpack-merge@ 5.7.3 installed at node_modules\_webpack-merge@5.8.0@webpack-merge
[11/12] rechoir@ 0.7.0 installed at node_modules\rechoir@0.7.1@rechoir
[12/12] import-local@ 3.0.2 installed at node_modules\_import-local@3.1.0@import-local
All packages installed (49 packages installed from npm registry, used 690ms(network 675ms), speed 713.3KB/s, json 49(165.8KB), tarball 315.67KB, manifests cache hit 0, etag hit 0 / miss 0)
[webpack-cli@4.9.2] link E:\devTools\nodejs\node_global\webpack-cli@ -> E:\devTools\nodejs\node_global\node_modules\webpack-cli\bin\cli.js
C:\Users\ly>
```

### 3.检查是否安装成功

cmd命令行窗口输入如下命令：

```
▼ XML | 复制代码
1 webpack -v
```

如下图所示，webpack和webpack-cli已经安装成功。



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

C:\Users\ly>webpack -v
webpack: 4.42.0
webpack-cli: 4.9.2
webpack-dev-server not installed

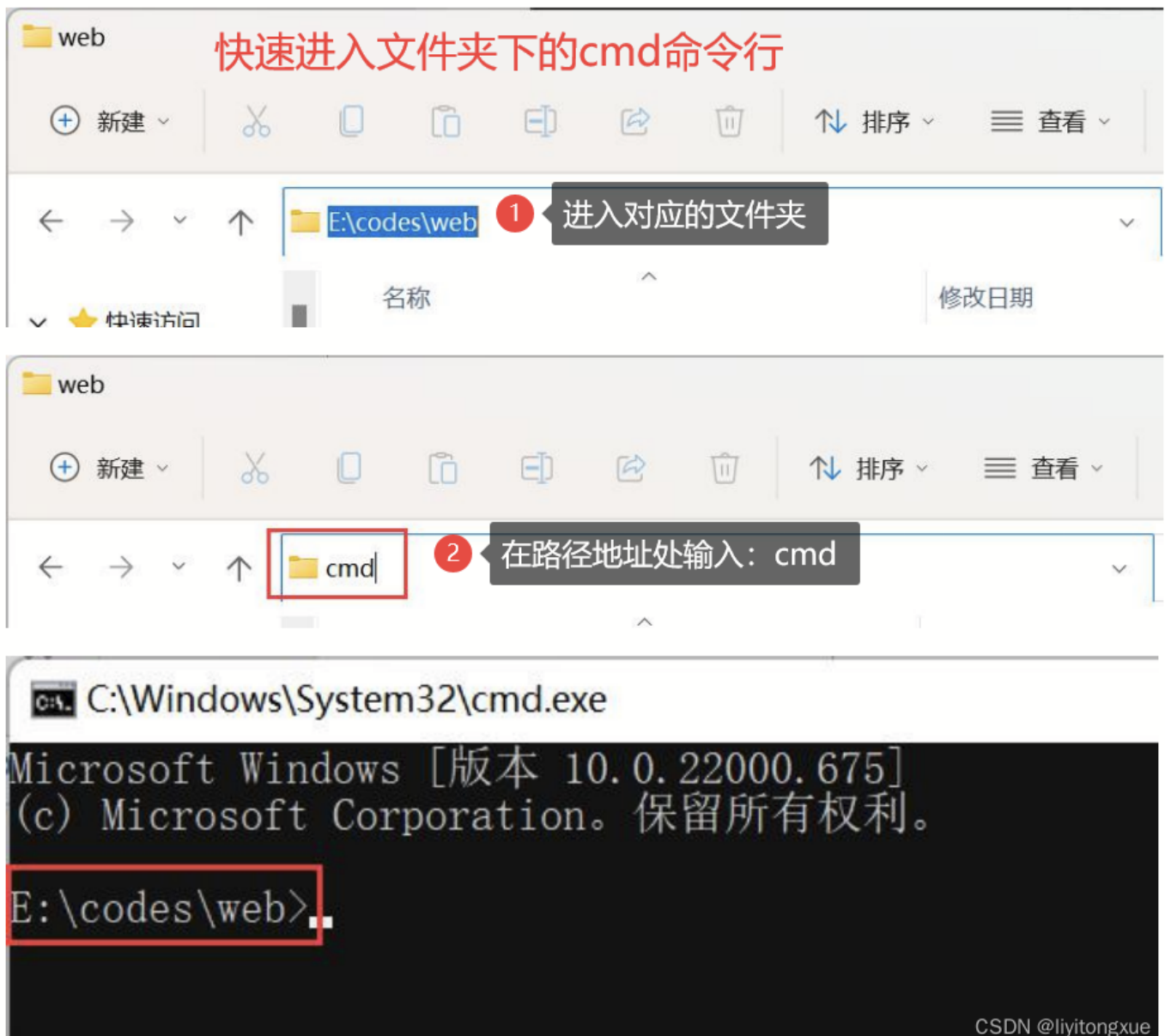
C:\Users\ly>_
```

CSDN @liyitongxue

## 8.新建一个默认的Vue项目

### 1.创建项目

首先进入【E:\codes\web】路径下（此处创建在E:\codes\web下，请根据自身实际情况更改），然后进入此文件夹的cmd命令行窗口。



执行如下命令创建一个名为【hello-vue】的Vue项目：

```
XML | 复制代码
```

```
1 vue create hello-vue
```

```
C:\> npm config get registry

Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation。保留所有权利。

E:\codes\web>vue create hello-vue

Vue CLI v5.0.4
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features

CSDN @liyitongxue
```

点击回车后，会自动下载所需文件（包括node\_modules），创建成功后如下图所示：

```
C:\Windows\System32\cmd.exe

Vue CLI v5.0.4
[ ] Creating project in E:\codes\web\hello-vue.
[ ] Installing CLI plugins. This might take a while...

added 843 packages in 15s
  [ ] Invoking generators...
  [ ] Installing additional dependencies...

added 95 packages in 3s
  [ ] Running completion hooks...

  [ ] Generating README.md...

  [ ] Successfully created project hello-vue.
  [ ] Get started with the following commands:

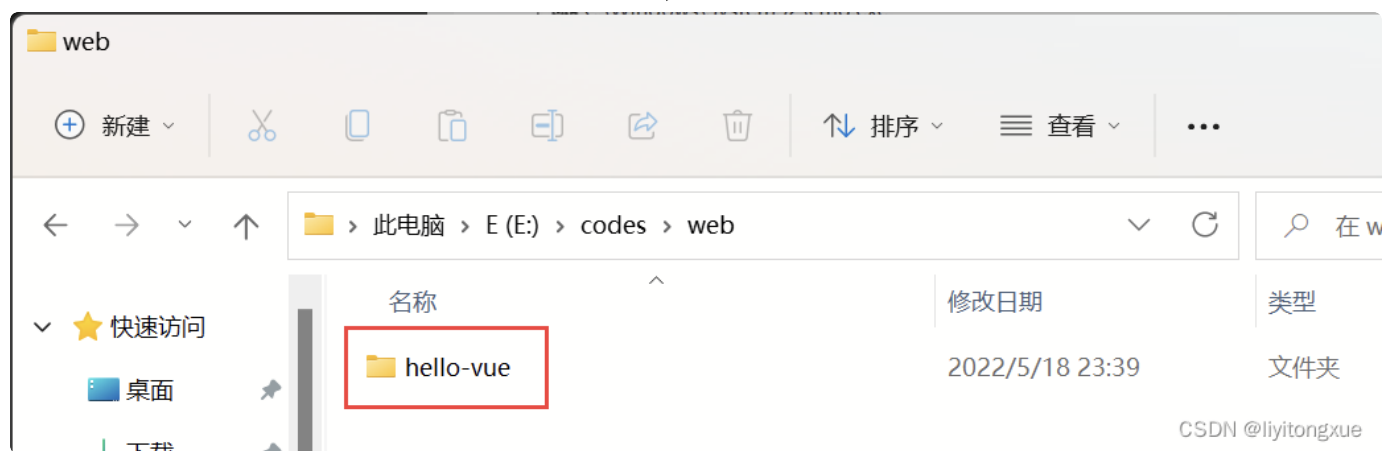
$ cd hello-vue
$ npm run serve

E:\codes\web>_

CSDN @liyitongxue
```

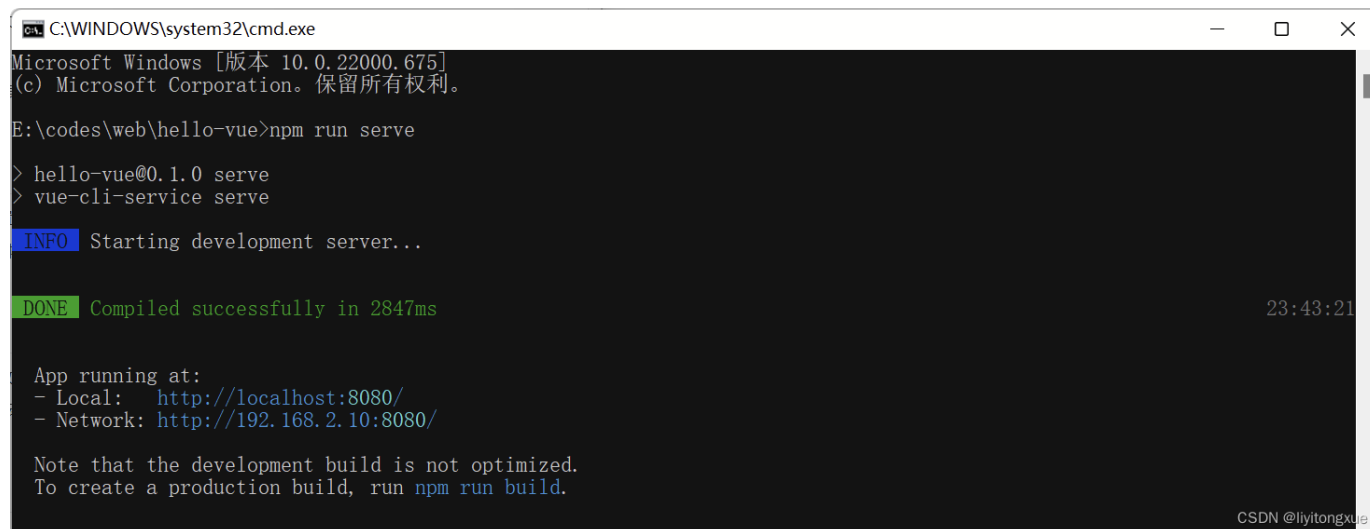
## 2.启动项目

首先进入【E:\codes\web\hello-vue】文件夹，然后进入此文件夹的cmd命令行窗口

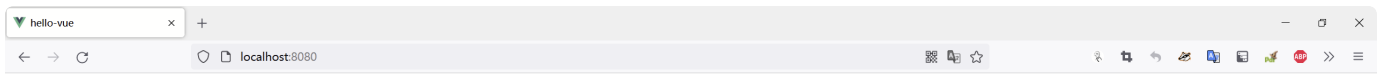


执行如下指令运行该项目：

```
1  npm run serve
```



如上图项目启动成功，在浏览器打开http://localhost:8080/进行访问，效果如下图所示：



## Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,  
check out the [vue-cli documentation](#).

### Installed CLI Plugins

[babel](#) [eslint](#)

### Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#) [News](#)

### Ecosystem

[vue-router](#) [vuex](#) [vue-devtools](#) [vue-loader](#) [awesome-vue](#)

CSDN @lylittong/vue